Camera models

September 16, 2020

1 Introduction

The aim of this short document is to show how a perspective camera can be formed. For this purpose, homogeneous coordinates has to be inroduced first, then the application of camera intrinsic and extrinsic parameters is overviewed.

2 Homogeneous coordinates

In the 3D space, a point is represented by three (floating point) number: X, Y, and Z. A vector $\boldsymbol{x}$ represents this spatial points as

$$\boldsymbol{x} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The homogeneus representation of this point is

$$\boldsymbol{x_h} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Thus, four coordinates are used to represent a 3D point with three degrees of freedom. The additinal one degree enables to set the scale of the vector x_h . Let us denote this scale by α. Due to the scale ambiguity,

$$\boldsymbol{x_h} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \sim \begin{bmatrix} \alpha X \\ \alpha Y \\ \alpha Z \\ \alpha \end{bmatrix}$$

From homogeneuus to Cartesian.

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \sim \begin{bmatrix} X/W \\ Y/W \\ Z/W \\ 1 \end{bmatrix} \rightarrow x = \begin{bmatrix} X/W \\ Y/W \\ Z/W \end{bmatrix}$$

2.1 Homogeneous → Descartes

It the homogeneous representation of a point is given the cartesian coordinates can be easily computed by dividing the elements of the vector by the last coordinate as

$$\boldsymbol{x_h} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \sim \begin{bmatrix} X/W \\ Y/W \\ Z/W \\ 1 \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} X/W \\ Y/W \\ Z/W \end{bmatrix}$$

3 Affine Transformations

The affine tansformations are the transformations for which the parallel lines remain parallel ones. They can be always written in the form

$$T_{aff} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

There are two important congruent transformation that is widely applied in computer vision: the rotation and the translation

3.1 Translation

If the Cartesian representation is transformed into a a homogeneous form,

$$\boldsymbol{T_{tr}} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If one multiplies a vector, given in homogeneous form, by $\boldsymbol{T}_{tr}$, it transforms the corrdinates as

$$\begin{bmatrix} X_i + d_x \\ Y_i + d_y \\ Z_i + d_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}.$$

3.2 Rotation

Objects in our 3D word are usually moving and rotating. For example, when a car is taking a turn, the chassis of the car is rotating around the vertical axis.

3.3 Rotation in 2D space

If a 2D point $[x_i \quad y_i]^T$ is given, and it is rotated around the origin of the system, the transformed coordinates x'_i and y'_i are obtained as

$$x'_i = x_i \cos\alpha - y_i \sin\alpha$$

$$y'_i = x_i \sin\alpha + y_i \cos\alpha$$

It can be written by a matrix as follows:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

3.4 Rotation in 3D

In the spatial domain, there are different representations for rotation. We use the simplest one: a 3D rotation is represented by three angles α, β, and γ. These angles represents the rotation around the X, Y, and Z axis, respectively. When the vectors to be rotated are represented by a homogeneus vector $[X_i \quad Y_i \quad Z_i \quad 1]^T$, the rotating matrices are as follows:

$$\boldsymbol{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R}_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R}_\gamma = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A very important property of this matrices is that they are orthonormal:

$$\boldsymbol{R}_\alpha^T \boldsymbol{R}_\alpha = \boldsymbol{R}_\alpha^T \boldsymbol{R}_\alpha = \boldsymbol{R}_\gamma^T \boldsymbol{R}_\gamma = \boldsymbol{I}$$

A general 3D rotation $\boldsymbol{R}$ can be written by three independent rotations around the principal axes:

$$\boldsymbol{R} = \boldsymbol{R}_{\boldsymbol{\gamma}} \boldsymbol{R}_{\boldsymbol{\beta}} \boldsymbol{R}_{\boldsymbol{\alpha}}$$

This rotation is also orthonormal as

$$(\boldsymbol{R_\gamma R_\beta R_\alpha})^T \boldsymbol{R_\gamma R_\beta R_\alpha} = \boldsymbol{R}_\alpha^T \boldsymbol{R}_\beta^T \boldsymbol{R}_\gamma^T \boldsymbol{R_\gamma R_\beta R_\alpha} = \boldsymbol{I}$$

The rotation matrix can be written as

$$\mathbf{T}_R = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix}$$

3.5 Rotation and translation

The concatenation of a translation and a rotation is determined by the multiplication of the rotation and translation matrices.

$$\mathbf{T}_R \boldsymbol{T}_{tr} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{d} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{Rd} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix}$$

If the translaton is the first transformation, and the rotation is the second one, it can be written that

$$\boldsymbol{T}_{tr} \mathbf{T}_R = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{d} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix}$$

The two different forms can be equlited:

$$\begin{bmatrix} \boldsymbol{R_1} & \boldsymbol{R_1 d_1} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_2} & \boldsymbol{d_2} \\ 0^{\boldsymbol{T}} & 1 \end{bmatrix}$$

In this case,
$\boldsymbol{R}_1 = \boldsymbol{R}_2$, and $\boldsymbol{R}_1 \boldsymbol{d}_1 = \boldsymbol{d}_2$, or $\boldsymbol{d}_1 = \boldsymbol{R}^T \boldsymbol{d}_2$.
A typical Euclidean transformation in 3D vision, as well as in computer graphics, the movement of a camera w.r.t. the world.

4 Projection

In computer vision, the following projection parameters are used:

- focal length f: the distance of the focal point and the camera image
- $[u_0 \quad v_0]^T$: principal point in the camera image. Principal points the location where the optical axis (axis Z) intersects the image plane.
- k_u and k_v are the horizontal and vertical sizes of the pixels, respectively. Their unit is pixel/m if the unit in the 3D worlds is meter.

This so-called intrinsic camera parameters are stacked in a matrix as follows:

$$K = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_u & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then the projection itself can be written by simply multiplíing the intrinsic matrix and a spatial point as

$$K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} fk_u X + Z u_0 \\ fk_v Y + Z v_0 \\ Z \end{bmatrix}$$

Then the homogeneous division should be carried out:

$$\begin{bmatrix} fk_u X + Z u_0 \\ fk_v Y + Z v_0 \\ Z \end{bmatrix} \sim \begin{bmatrix} fk_u \frac{X}{Z} + u_0 \\ fk_v \frac{Y}{Z} + v_0 \\ 1 \end{bmatrix}$$

This is equivalent to the projection equation, the only difference here is that offset $[u_0 \quad v_0]^T$ is added to the projected locations.